

# GA-Hard Functions Built by Combination of Trap Functions

Manuel Clergue, Philippe Collard

I3S Laboratory, University of Nice Sophia-Antipolis

Les Algorithmes/Euclide, 2000 route des Lucioles, Biot Sophia Antipolis, F-06410

{pc,clerguem}@i3s.unice.fr

**Abstract - We propose to construct hard functions for genetic algorithms by combining two types of misleading functions. We consider on one hand the traditional Trap functions defined over the unitation, and on the other hand new Trap functions based on the alternation. We recall the performances of GA on these functions as well as the results on the predictive value of the coefficients of correlation between distance to the optimum and fitness. We show that the combination of such functions can generate misleading problems for a genetic algorithm. Moreover, some of these combinations constitute counterexamples for the predictive value of the coefficient of correlation.**

## I. Introduction

A central problem consists in characterizing the functions which are difficult to optimize with a genetic algorithm. Many work is based on the concept of deception, defined relatively to the average fitness of the schema in competition [5], [6], [10]. Misleading functions, the so-called *Trap functions*, were directly built by synthesizing their plot in the distance to the optimum/fitness plan in the form of a piecewise linear function. We start by pointing out the traditional definition of these functions based on the Hamming distance, then we extend this family of functions by considering a new distance more in connection with the cross-over operator. Then, we propose to synthesize new functions by linear combination of Trap functions. Lastly, we show that this combination can be used to build counterexamples concerning the predictive value of the coefficient of correlation between the distance to the optimum and the fitness, the FDC.

## II. Trap Functions

Trap functions represent a well-known family of functions conceived to be misleading for a genetic algorithm. Such functions exhibit a local optimum towards which the population converges.

### A. U-Trap: Trap on Unitation

#### A.1 Definition

U-Trap functions are directly characterized by a linear piecewise function defined in the unitation/fitness plan. The search space is divided in two basins of attraction in the Hamming space, leading respectively towards the global optimum and the local one. These two optima are represented by complementary bit strings. We will suppose in the following, without loss of generality, that 0..0 represents the global optimum and 1..1 the local one. In this case, the unitation of a string (the number of 1) represents its Hamming distance from the global optimum. Two real parameters in the interval  $[0,1]$ ,  $b$  and  $r$ , characterize a U-Trap:  $b$  represents the relative width of the basins of attraction, whereas  $r$  measures the ratio of fitness between the two optima. The function U-Trap is defined by:

$$\text{U-Trap}(x) = \begin{cases} 1.0 - \frac{u(x)}{b} & \text{if } u(x) \leq b \\ \frac{r(u(x)-b)}{1.0-b} & \text{elsewhere} \end{cases}$$

where  $u(x)$  is the normalized unitation (i.e. divided by strings length).

Using the schema theory, Deb and al. [5], gave theoretical conditions on  $b$  and  $r$  to make the function U-Trap completely or partially misleading. Under these conditions, the algorithm is misled while being “trapped” in the basin of attraction of the local optimum.

#### A.2 Unitation/Fitness Correlation

Some researchers established a correspondence between the evolutionary algorithms (EA) and the heuristic of search in a space of state [6]. The fitness landscape in the EA corresponds to the space of the states and the fitness to the search heuristics. In the majority of search heuristics, like algorithm A\*, or Means-Ends Analyze, the label is interpreted like a measurement of the distance to the goal. There are many results which show that the better it represents a good estimate to the goal, the better the heuristic is. On the basis of these remarks,

Jones [6] suggested that the ideal fitness function for a EA should give an indication of the distance which it remains to reach the optimum. Thus, he proposed to use the correlation between the fitness and the distance to the nearest optimum to measure the difficulty of a fitness landscape for a genetic algorithm.

The definition of the correlation unitation/fitness (FUC, in the following) is quite simple. Given a set  $F = \{f_1, f_2, \dots, f_n\}$  of the fitness values of  $n$  individuals and the  $U = \{u_1, u_2, \dots, u_n\}$  correspondent distances to the optimum, the correlation is calculated by:

$$r = \frac{c_{FU}}{\sigma_F \sigma_U}, \text{ where } c_{FU} = \frac{1}{n} \sum_{i=1}^n (f_i - \bar{f})(u_i - \bar{u})$$

is the covariance between  $F$  and  $U$ , and  $\sigma_F$ ,  $\sigma_U$ ,  $\bar{f}$  and  $\bar{u}$  are respectively the standard deviation and the mean of  $F$  and  $U$ . Problems should be divided in 3 classes, according to the value of their FUC:

1. deceptive problems ( $\text{FUC} > 0.15$ ) : the fitness gives wrong information about distance to optimum
2. hard problems ( $-0.15 < \text{FUC} < 0.15$ ) : the fitness gives no information about distance to optimum
3. easy problems ( $\text{FUC} < -0.15$ ) : the fitness gives correct information about distance to optimum

The second class corresponds in fact to problems for which difficulty cannot be predicted : the value of the FUC brings little information about the structure of the landscape. In this case, Jones recommends to trace the scatter plot of the function; he claims that this is useful to distinguish, for example, a *needle in a haystack* problem from a symmetrical problem. Both have a null FUC, but the first one is hard for GA (for all methods, indeed), and some instances of the second may be easy.

The figure 1 shows the FUC values for Trap Functions according to  $r$  and  $b$  parameters. We notice that the easy and hard areas predicted by the FUC nearly correspond to those defined by Deb.

### A.3 Performances

We present now the performances of a GA facing to U-Trap functions. The GA is a simple GA with standard settings. The performance is defined as the number of runs which hits the global optimum in less than 200 generations divided by the total number of runs (50). The chromosomes length is 26 bits, the population size is set to 50 and we use a tournament selection without elitism. Figure 2 represents the performance according to  $r$  and  $b$  parameters. We notice that the actual difficult area (where the performance is below 0.05) nearly corresponds to the one that FUC predicted hard.

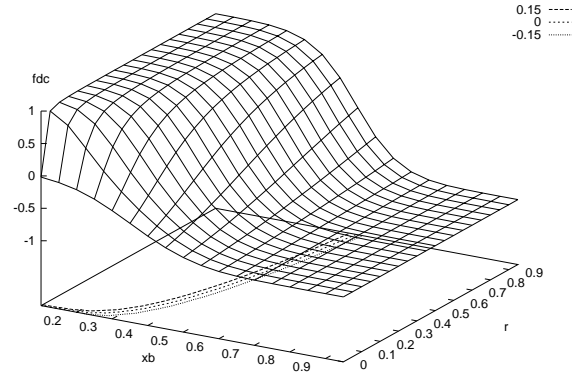


Fig. 1. Correlation coefficient FUC for the U-Trap functions according to  $b$  and  $r$  parameters. The isolines allow to separate the easy, hard and deceptive areas according to Jones classification.

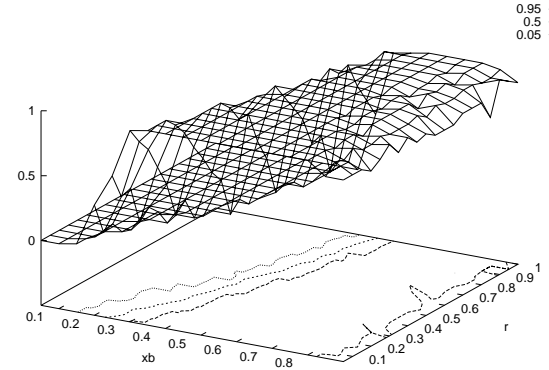


Fig. 2. Performances of the GA for U-Trap functions according to  $b$  and  $r$  parameters.

### B. Trap Functions on Alternation

Some studies address the construction of function over alternation. Culberson [4] showed that its crossover based GA, GIGA, performs better than a standard GA on maximizing alternation. He explained this by exhibiting an isomorphism between mutation space and crossover space. According to him, the alternation has the same role for crossover than unitation for mutation. Naudts and Naudts [7] introduce the Ising Model, which can be seen as alternation function, in order to explain the negative effect of symmetry on the convergence on the simple genetic algorithm.

In this section, we consider trap functions designed from alternation. These functions were first proposed by

Collard and Clergue [2] in order to point out a new kind of GA deceptiveness, more related to crossover.

### B.1 Definition

The so-called A-Trap functions are designed in the same way that the U-Trap, except that alternation is used instead of unitation. More precisely, A-Trap functions are defined as follow:

$$\text{A-Trap}(x) = \begin{cases} 1.0 - \frac{a(x)}{x_b} & \text{if } a(x) \leq x_b \\ \frac{r(a(x) - x_b)}{1.0 - x_b} & \text{elsewhere} \end{cases}$$

where  $a(x)$  is the number of alternation divided by  $\lambda - 1$ , the maximum number of alternation in a bit string of length  $\lambda$ . The search space is divided in two basins of attractions; A-Trap functions admit as optima two pairs of complementary strings, the global one (00..0, 111..1) and the local one (0101.., 1010..). Let us notice that, opposite to the case of U-Trap functions the global optima and the local ones are not complementary strings. Instead, the Hamming distance between global optima and local ones is  $\lambda/2$ , if the length of the strings is  $\lambda$ .

There is a inherent symmetry for A-Trap functions: two complementary strings have the same fitness. As a consequence, their FUC is null, so it cannot predict the difficulty of the function.

### B.2 alternation / fitness correlation

An open problem is to characterize the types of fitness landscapes for which crossover will be an effective operator. Altenberg ([1]) proposed to use the number of alternations (discontinuities) as a measure of *crossover* distance between solutions. His argument is based on the following fact: considering a sequence of crossover events on complementary strings that produce a path to the optimum, as one moves further from the optimum along the path, the number of alternations increases by one with each step. So, the number of single-point crossovers needed to transform a complementary pair into the global optimum and its complement can be used as a measure of difficulty to move through the search space by crossing over.

Following Altenberg, we define the FAC, the correlation coefficient between alternation and fitness, in the same way the FUC is defined. By construction, the FAC of A-Trap function is equal to the FUC of U-Trap function (see [2] for details).

### B.3 Performances

We evaluate the performances of GA on A-Trap functions with the same settings we used for U-Trap functions. The figure 3 shows the variation of the performance according to the value of  $b$  and  $r$ . Putting this

figure next to the FUC one (figure 1, to be interpreted here as the FAC in function of  $b$  and  $r$ ), we can see that the effective misleading zone is nearly the same than the predicted one. In this case, the FAC removes the ambiguity due to the null FUC.

In the easy area, it appears that the GA is not completely successful. The performance rate is around 0.9, i.e. 10% of runs fail to find one of the global optima. Deeper analyze shows that in fact the GA converges close to the optimum, but is unable to make the last steps to success.

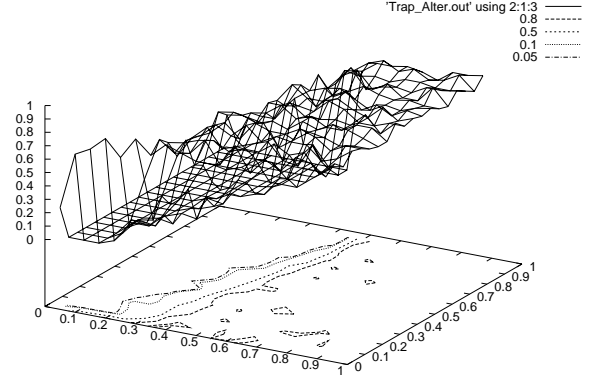


Fig. 3. Performances of the GA for A-Trap functions according to  $b$  and  $r$  parameters.

## III. How to Combine A-Trap and U-Trap

In this section we define the so-called AU-Trap functions as new deceptive functions based on U-Trap and A-Trap functions. According to the parameter  $b$ , we present two ways to combine U-Trap with A-Trap functions.

### A. Definition

#### A.1 Type I AU-Trap function

We consider the combination between one U-Trap and one A-Trap based on the same piecewise linear function, that is with the same parameters  $b$  and  $r$ . So,

$$\text{AU-Trap1}(b, r) = \alpha \cdot \text{A-Trap}(b, r) + (1 - \alpha) \cdot \text{U-Trap}(b, r)$$

where  $\alpha$  is a real parameter in the range  $[0, 1]$ . According to the fitness distance correlation coefficient, FAC for A-Trap and FUC for U-Trap, difficulty prediction is the same for the two functions. So, type I combines two functions with the same level of difficulty but different nature of hardness.

## A.2 Type II AU-Trap function

For type II functions, we build combination based on opposite piecewise linear function. To do this, we just inverse the basin of attraction in such a way that functions difficulties become opposite: when the U-Trap is easy, the A-Trap is hard and vice versa. So,

$$\text{AU-Trap2}(b, r) = \alpha \cdot \text{A-Trap}(b, r) + (1 - \alpha) \cdot \text{U-Trap}(1 - b, r)$$

where  $\alpha$  is a real parameter in the range  $[0, 1]$ .

## B. AU-Trap functions and fitness distance correlation

Now, we have a look for the correlation coefficients, FUC and FAC, for type I and type II AU-Trap functions. The parameter  $r$  is set to 0.9,  $b$  and  $\alpha$  vary between 0 and 1. U-Trap functions (resp. A-Trap) are a special case with FUC varying from  $-1$  to  $+1$  and quasi null FAC (resp. FAC varying from  $-1$  to  $+1$  and null FUC). In the space FUC/FAC, these two classes of functions are located on the horizontal axe and the vertical one respectively. Type I AU-Trap functions (resp. type II) are situated in northeast or southwest sector (resp. northwest or southeast); the product FUC.FAC is positive (resp. negative). During the transitions between type I and type II, there is a continuity for correlation coefficients (see fig. 4). (FUC, FAC) points are roughly located on concentric circles. We can observe that variations of  $\alpha$  (i.e. weighting between U-Trap and A-Trap) correspond to moves on an arc, while variations of parameter  $b$  (i.e. difficulty of the trap) correspond to moves on a diameter.

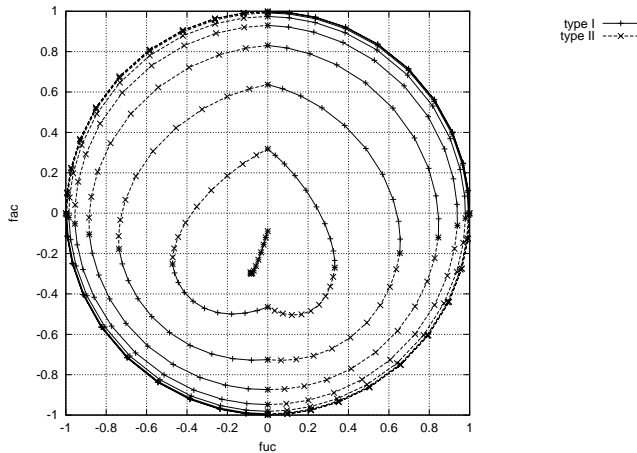


Fig. 4. Fitness distance correlation in the space FUC/ FAC for the class of AU-Trap functions. Points on a same diameter correspond to functions with the same parameter  $b$ .

## C. Performances of AU-Trap functions

We study the performance of a simple genetic algorithm, with standard setting, applied on AU-Trap functions. We use chromosome of length 26, one population of size 200 and roulette wheel selection with elitism. The crossover rate is set to 0.7 and the probability of mutation is 1 per chromosome. Performance is defined as the number of runs for which the global optimum is reached in less than 500 generations divided by the total number of runs (20). Figure 5 (resp. Figure 6) shows the performance of AU-Trap type I functions (resp. type II) in the space (FUC, FAC).

We can observe on figure 5 that an AU-Trap type I function, for which there is no conflict between the difficulty of each component ( $\text{FUC.FAC} > 0$ ), have performances consistent with the ones predicting by the couple (FUC, FAC). An AU-Trap function with a positive couple (FUC, FAC) (resp. negative) is misleading (resp. easy).

AU-Trap type II functions are more interesting to study because there is a conflict between prediction values according to the FUC or the FAC ( $\text{FUC.FAC} < 0$ ). On the other hand (see figure 6) it is difficult to interpret results. Globally, one observe that the negative correlation coefficient is the more significant in order to predict deception. One can remark that performance of A-Trap functions (cf. the line  $\text{FUC} = 0$ ) presents a curious behaviour compared to closely related functions in the space FUC/FAC.

In order to deeply analyse the performances of AU-Trap type II functions, we focus our study on specific functions from cross-section in the space FUC/FAC/performances according to a diameter: the  $\alpha$  parameter is set to 0.6, and the  $b$  parameter varies from 0.1 to 0.9. Figure 7 shows the evolution of FUC and FAC, and the performance, according to parameter  $b$ . Globally, we observe that only one of the two coefficients have a predictive value according to  $b$ ; this shows that this class of functions is a potential spring of counterexamples for each coefficient of correlation.

## D. Counter-examples for FDC

There are many works about the search of counterexample for FUC [1], [3], [9], [8]. According to Altenberg, the fact that FUC is only a statistical and static measure, based on a distance which is apparently only bound to mutation, implies two assumptions: either Hamming distance is connected to the way genetic algorithms work, or this relation exists in a fortuitous way among the test set chosen by Jones. In which case, counterexamples exists for which this relation does not hold, and which, therefore, deceive the FUC. Since it seems that there is

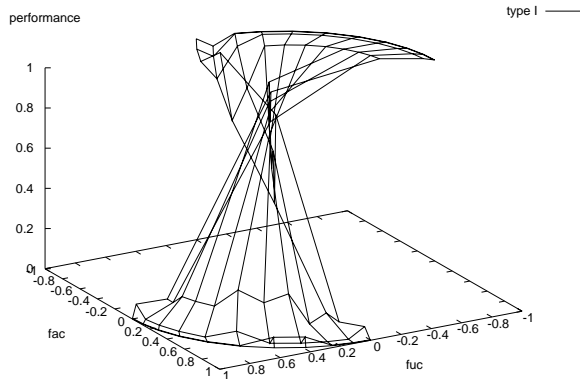


Fig. 5. Performances of a simple genetic algorithm on the class of AU-Trap type I functions according to the correlation coefficients FUC and FAC.

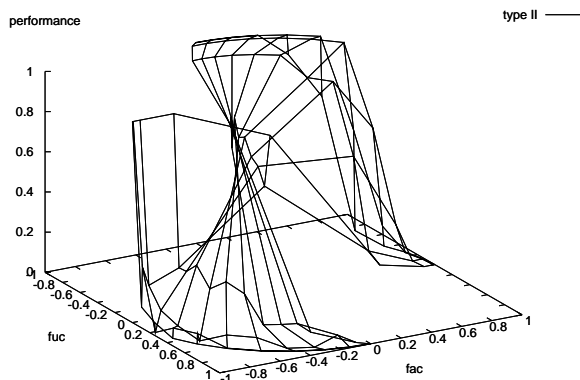


Fig. 6. Performances of a simple genetic algorithm on the class of AU-Trap type II functions according to the correlation coefficients FUC and FAC.

no relation between recombination operators and Hamming distance, and that mutation is supposed to play a marginal role in genetic algorithms, Altenberg claims that it is possible to construct a counterexample. The one he constructs is GA-easy, but the FUC is null ; further, the observation of the scatter plots gives no more information [1]. Quick et al. construct easy functions (ridges functions) with a high positive FUC [9], [8].

While the Altenberg's counter example is prone to discussion, in particular on the definition of GA-easiness, the one of Quick et al. is clear: there are functions that the FUC predicts misleading and which are in fact easy. Nevertheless, all these counter examples exploit known weaknesses of the FUC: its nullity for symmetrical functions or the low contribution of a particular path in the

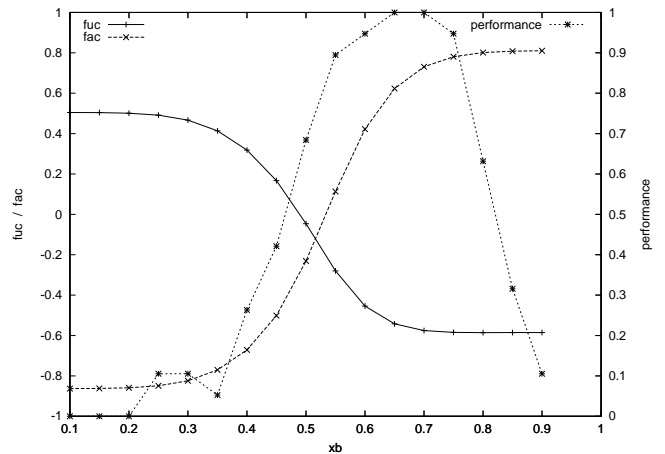


Fig. 7. Performances of a simple genetic algorithm on the class of AU-Trap type II functions according to the parameter  $b$  ( $\alpha$  is set to 0.6).

global calculation. Beside, Quick et al. recognize that the FUC calculated with the points actually sampled by the GA gives better results.

#### D.1 Fitness/unitation correlation coefficient

We present, as counter examples for the FUC, GA-deceptive functions with a high negative FUC. Let us take into account AU-Trap type II functions for which the weight  $\alpha$  is set to 0.6, and the parameter  $b$  varies from 0.1 to 0.9 (cf. figure 7). For the values of  $b$  higher than 0.85, the function is deceptive (performances equal to 0.3 and 0.1). The FAC (+0.80) is a good predictor while the FUC (-0.6) provides a bad prediction. So, these functions are counterexamples for the predictive value of the fitness/unitation correlation coefficient.

#### D.2 Fitness/alternation correlation coefficient

We present, as counterexamples for the FAC, functions GA-deceptive with a high negative value FAC. Let us consider the class of AU-Trap type II functions with  $\alpha$  set to 0.6 and  $b$  lower than 0.35. These functions are GA-deceptive (performances  $< 0.1$ ). The FUC is a good predictor ( $0.4 < \text{FUC} < 0.5$ ) while the FAC (FAC  $< -0.7$ ) gives bad prediction. These functions are counterexamples for the FAC.

## IV. Conclusion

In this paper we have proposed a new class of GA-hard functions. These functions are designed from deceptive trap functions. We have considered two types of trap functions. First, the well-known U-Trap functions, based on Hamming distance, for which difficulty relies rather on mutation ; second, the A-Trap functions associated

to the crossover. At each class of functions, corresponds its own type of deceptivity. We have shown that each type can be predict by its own fitness distance correlation coefficient. Combining these two kind of difficulty in a same function allows to construct GA-hard functions and to exhibit counter examples for the correlation coefficient.

In future works, it will be usefull to extend our studies to the role of genetic operators on the performances of U-Trap and A-Trap functions, and their influence on the predictive power of FUC and FAC.

## References

- [1] L. Altenberg. Fitness Distance Correlation Analysis: An Instructive Counterexample. In Thomas Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97)*. Morgan Kaufmann, 1997.
- [2] M. Clergue and P. Collard. Misleading functions for genetic algorithms, designed from alternation. In *IEEE International Congress on Evolutionary Computation*, pages 1056–1063, 2000.
- [3] P. Collard, A. Gaspar, M. Clergue, and C. Escazut. Fitness distance correlation, as statistical measure of genetic algorithm difficulty, revisited. In *Proceedings of the European Conference on Artificial Intelligence*, pages 650–654. John Wiley & Sons, Ltd, 98.
- [4] Joseph C. Culberson. Mutation-crossover isomorphisms and the construction of discriminating functions. *Evolutionary Computation*, 2(3):279–311, 1994.
- [5] K. Deb and D. E. Goldberg. Analysing deception in trap functions. In L. D. Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 93–108. Morgan Kaufmann, 1993.
- [6] T. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In L. Eshelman, editor, *ICGA’95: Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 184–192. Morgan Kaufmann, 1995.
- [7] Bart Naudts and Jan Naudts. The effect of spin-flip symmetry on the performance of the simple GA. In Agoston E. Eiben, Thomas Bäck, Marc Schoenauer, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature – PPSN V*, pages 67–76, Berlin, 1998. Springer.
- [8] R.J. Quick, V.J. Rayward-Smith, and G.D. Smith. Fitness Distance Correlation and Ridge Functions. In *Proceedings of the fifth Conference on Parallel Problems Solving from Nature*, 1998.
- [9] R.J. Quick, V.J. Rayward-Smith, and G.D. Smith. Ridge Functions. Technical report, University of East Anglia, 1998.
- [10] L. D. Whitley. Fundamental principles of deception in genetic search. In G. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 221–241. Morgan Kaufmann, San Mateo, CA, 1991.